Recent Developments in Overture

Bill Henshaw

Centre for Applied Scientific Computing,
Lawrence Livermore National Laboratory,
Livermore, CA, USA 94551

www.llnl.gov/casc/Overture

9th Overset Grid Symposium, Penn State University, Pennsylvania
October 2008. LLNL-PRES-407076.

Background: Schlieren image from a detonation hitting a collection of moving rigid cylinders.

Acknowledgments

Supported by

Department of Energy, Office of Science

MICS Program: Mathematical, Information, and Computational Sciences

SciDAC: Scientific Discovery through Advanced Computing

LLNL: Laboratory Directed Research and Development (LDRD) program.

Current Overture developers

Kyle Chand

Bill Henshaw

Introduction

NEW Overture Version 22 is the first parallel version of Overture.

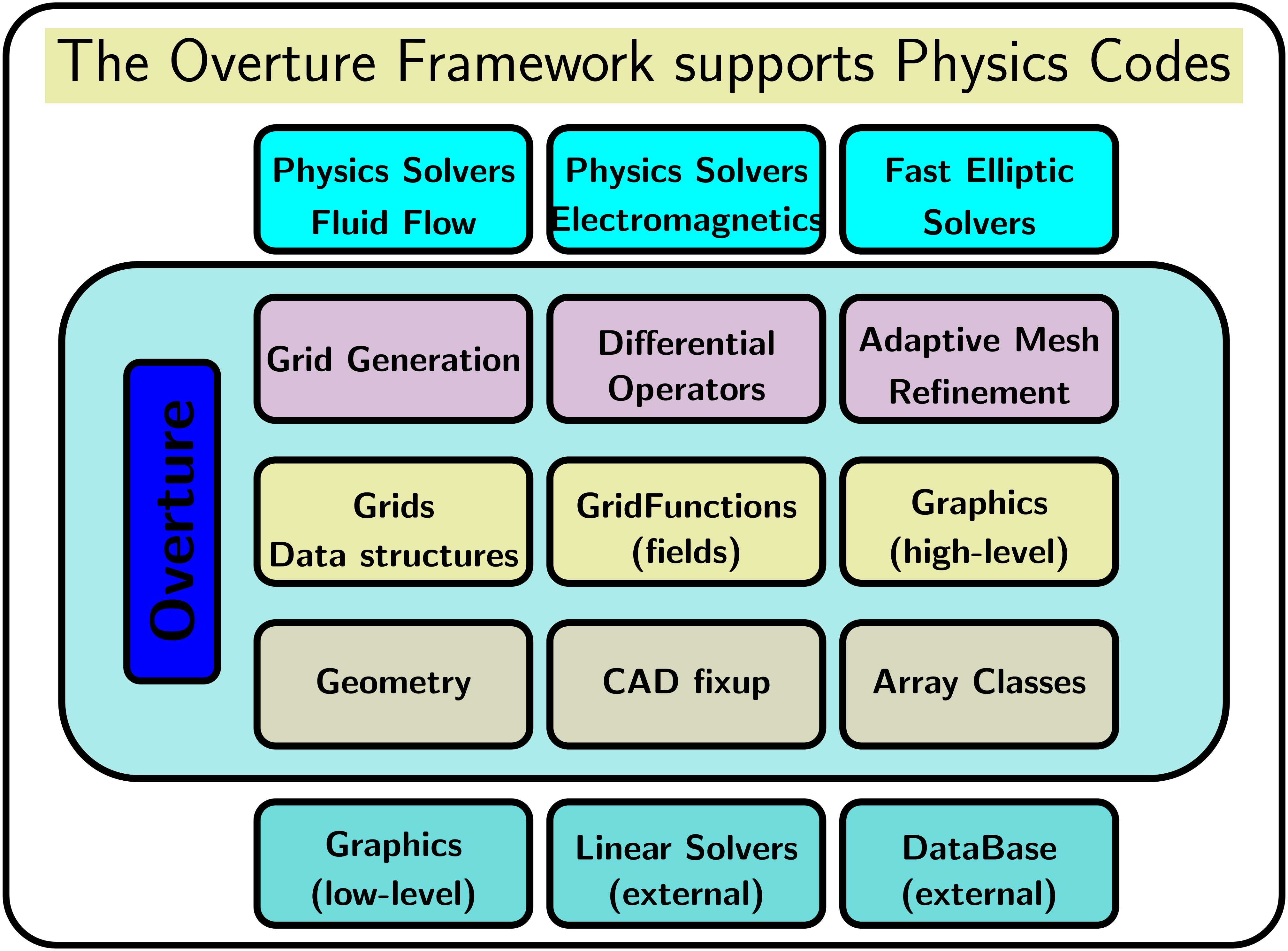
NEW The CG (Composite-Grid) Suite of PDE solvers has been released.

This talk will give an overview of some of the new capabilities including parallel capabilities:

- cgins : incompressible flow.
- cgcns : compressible flow with adaptive mesh refinement.
- ogen : overlapping grid generation.
- cgmx : Maxwell's equations.
- cgmp : multi-physics problems (see Kyle Chand's presentation).

Other notes of interest:

- interactive parallel graphics is working.
- parallel I/O is working (getting around the limitations of HDF5).


The Overture Framework supports Physics Codes
Physics Solvers Fluid Flow
Physics Solvers Electromagnetics
Fast Elliptic Solvers
Overture
Grid Generation
Differential Operators
Adaptive Mesh Refinement
Grids Data structures
GridFunctions (fields)
Graphics (high-level)
Geometry
CAD fixup
Array Classes
Graphics (low-level)
Linear Solvers (external)
DataBase (external)

Overture is toolkit for solving partial differential equations on structured, overlapping and hybrid grids.

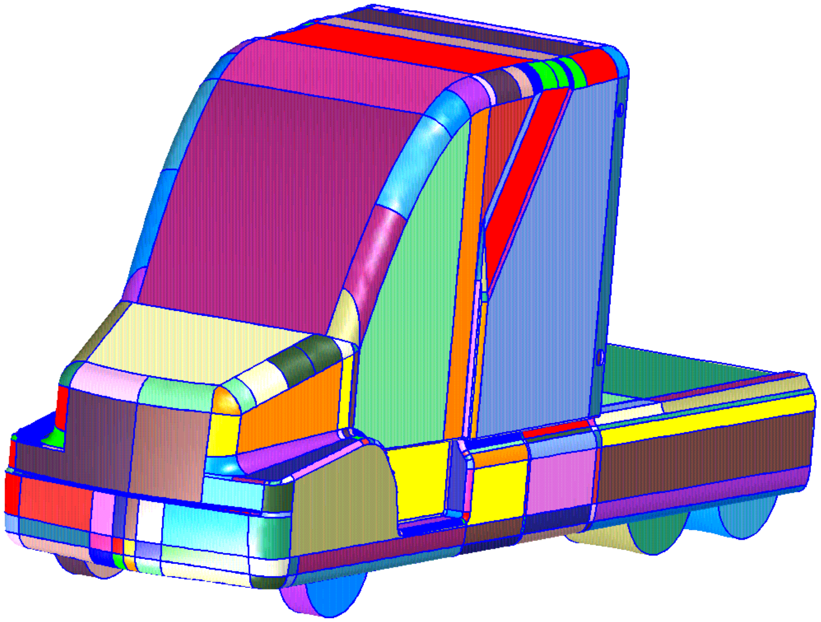
Key features:

- provides a high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- provides a library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (now using HDF5).
- command scripts with embedded perl commands.
- PDE solvers built upon Overture include those from the CG suite:
  - cgins: incompressible Navier-Stokes with heat transfer.
  - cgcns: compressible Navier-Stokes, reactive Euler equations.
  - cgmx: time domain Maxwell's equations solver.
  - cgmp: multi-physics solver (e.g. thermal hydraulics)

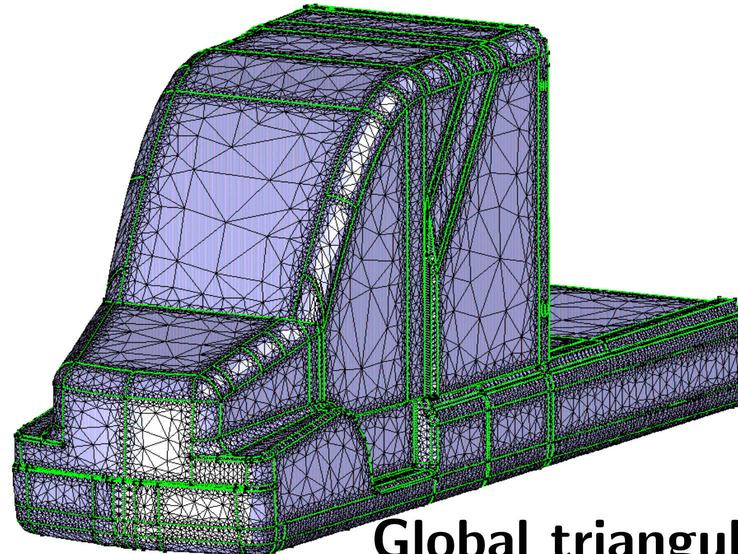
Overture supports a high-level C++ interface (but is built mainly upon Fortran kernels):

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

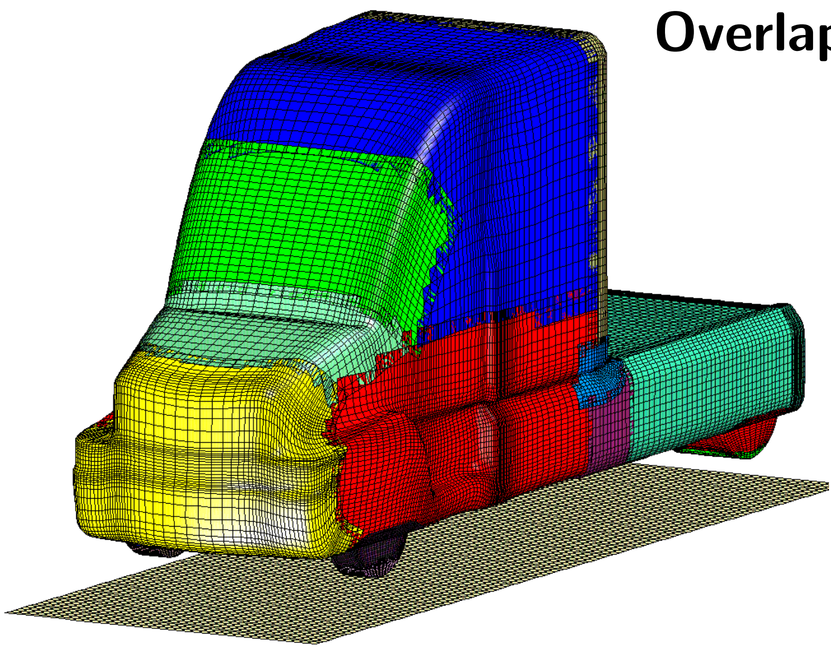
```
CompositeGrid cg;  // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg);  // create a grid function
u=1.;
CompositeGridOperators op(cg);  // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
  u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
  t+=dt;
  u.interpolate();
  u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
  u.finishBoundaryConditions();
}
```



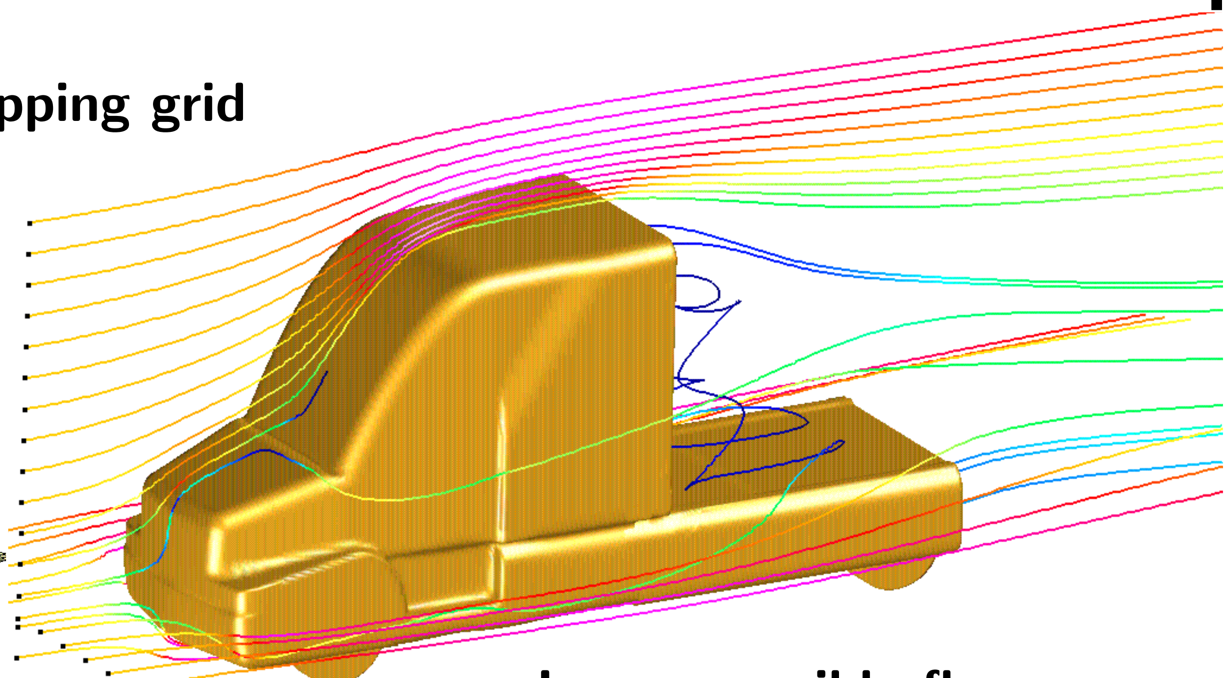
Cad geometry



Global triangulation



Overlapping grid



Incompressible flow.

Other Projects Using Overture (items not covered in this talk)

- ♢ Compressible multiphase flows (Don Schwendeman, RPI)
- ♢ Compressible multi-material flows and FCT algorithms (Jeff Banks, LLNL)
- ♢ Compressible axisymmetric flow with swirl (Kyle Chand, LLNL)
- ♢ Compressible flow/ice-formation (Graeme Leese, Nikos Nikiforakis, U. Cambridge).
- ♢ Einstein field equations (Philip Blakely, U. Cambridge).
- ♢ Converging shock waves with obstacles (Veronica Eliasson, Caltech).
- ♢ Blood flow in veins with obstacles. (Mike Singer, LLNL).
- ♢ Visco-plastic flows and friction-stir welding (Graeme Thorn, U. of Liverpool).
- ♢ Flow on the surface of the eye (Kara Maki, U. Delaware).
- ♢ Elasticity. (Daniel Appëlo, Caltech).

Cgins: Incompressible Flow

Features:

• 2nd-order and 4th-order accurate incompressible Navier-Stokes† (DNS).

• support for moving rigid-bodies (not parallel yet).

• Heat transfer (Boussinesq approximation).

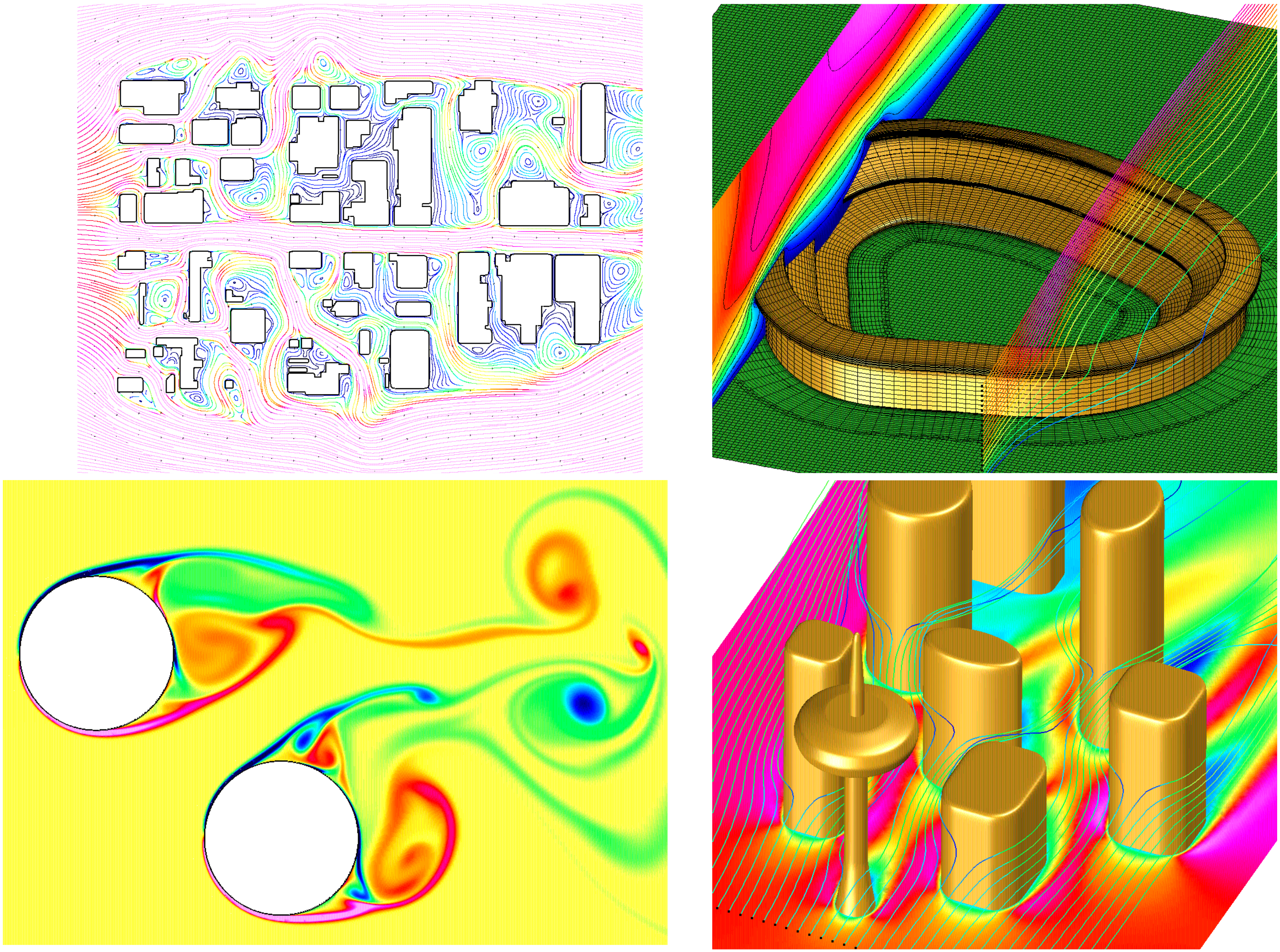
• semi-implicit (time accurate), pseudo steady-state (efficient line solver), full implicit (new).

• Visco-plastic model (full-implicit method).

• K-ϵ model (full-implicit method, in progress).

♢ Implicit systems and the pressure equation can be solved in parallel using PETSc, Hypre, SuperLU (and other packages available through the PETSc interface). Still under development is the parallel version of the multigrid solver Ogmg.

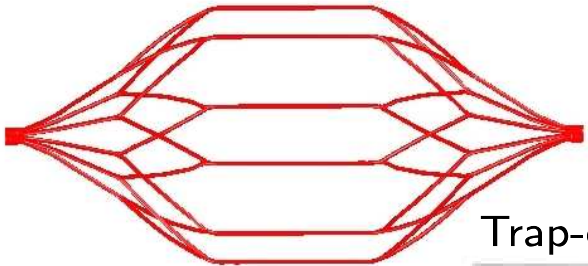
†Reference:
WDH., *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids*, J. Comput. Phys, **113**, no. 1, (1994) 13–25.



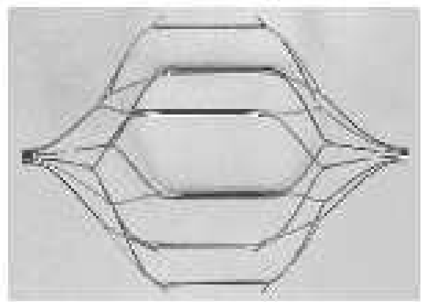
Incompressible flow computations with cgins.

Flow past a blood-clot filter using cgins (Mike Singer, LLNL)

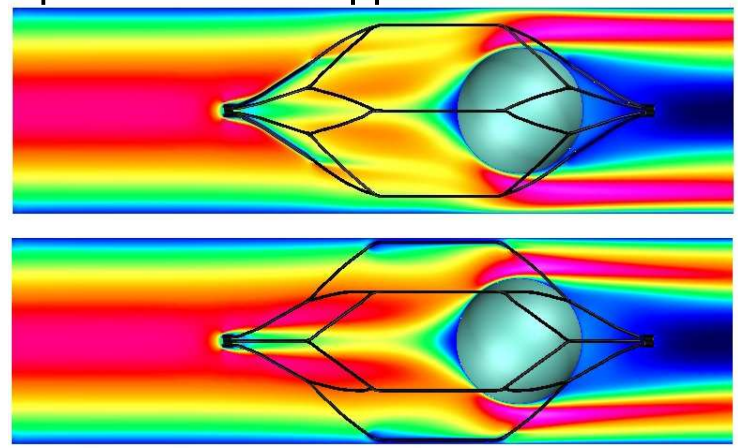
Overlapping grid for the filter

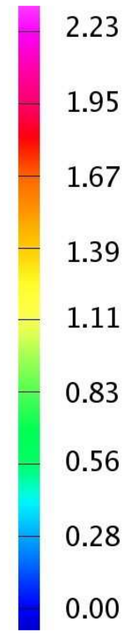


Trap-ease wire filter

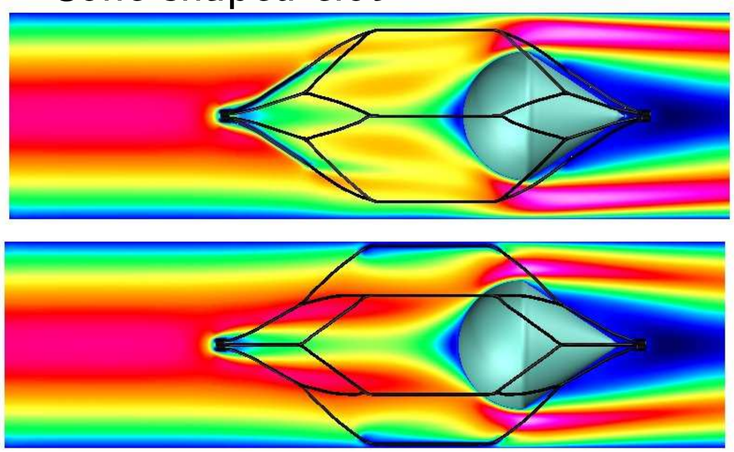


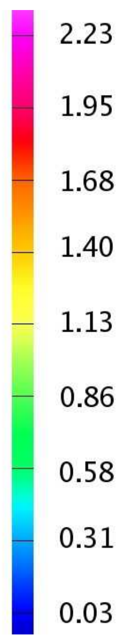
Spherical clot trapped in the filter



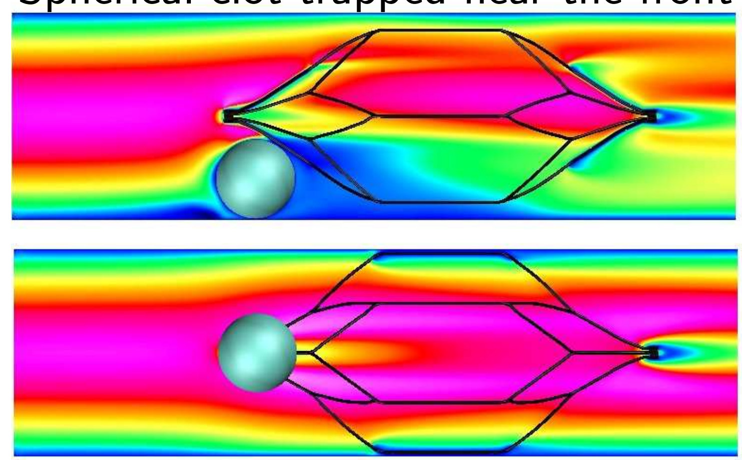


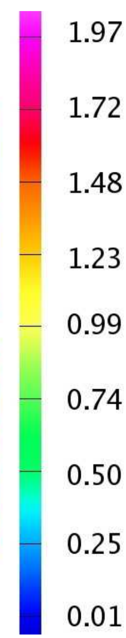
Cone shaped clot



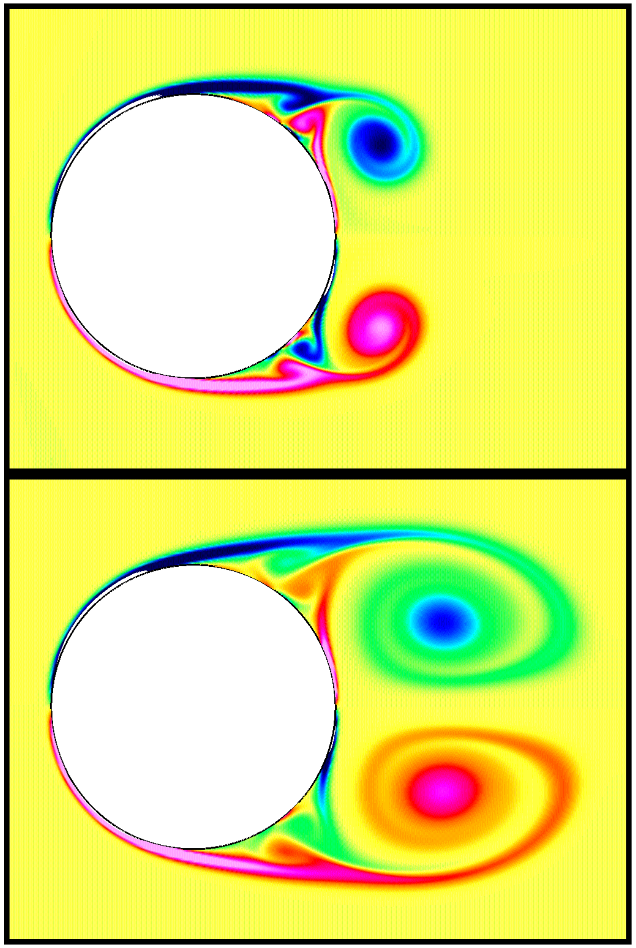


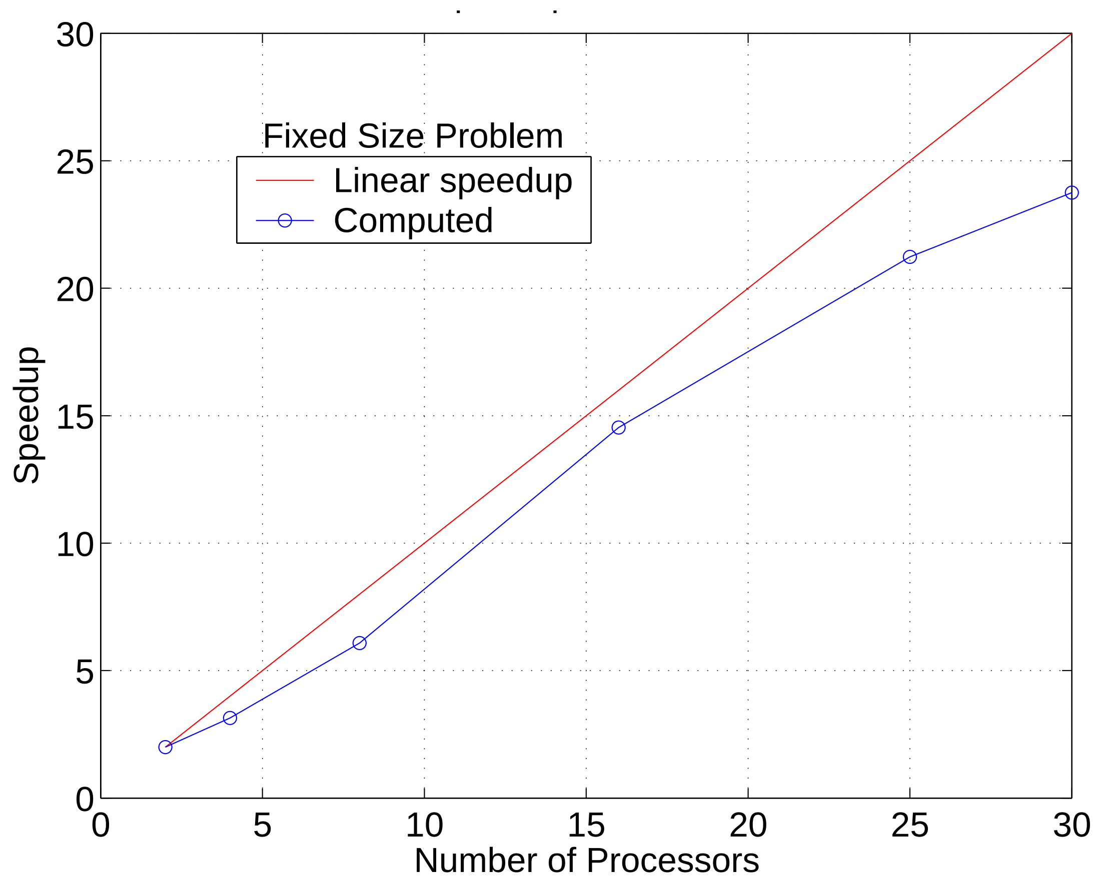
Spherical clot trapped near the front





Incompressible Navier-Stokes: preliminary parallel results





Left: impulsively started cylinder in an incompressible flow (vorticity). Right: parallel speedup keeping the problem size fixed (4 Million grid points), on a linux cluster (Xeon processors). The pressure equation is solved with algebraic multigrid (Hypre).

Parallel Adaptive Mesh Refinement on Overlapping Grids

♢ AMR: adaptive grids are dynamically created every few time-steps based on an estimate of the error.

♢ Parallel: Each base grid or refinement grid can be independently distributed over a set of processors.

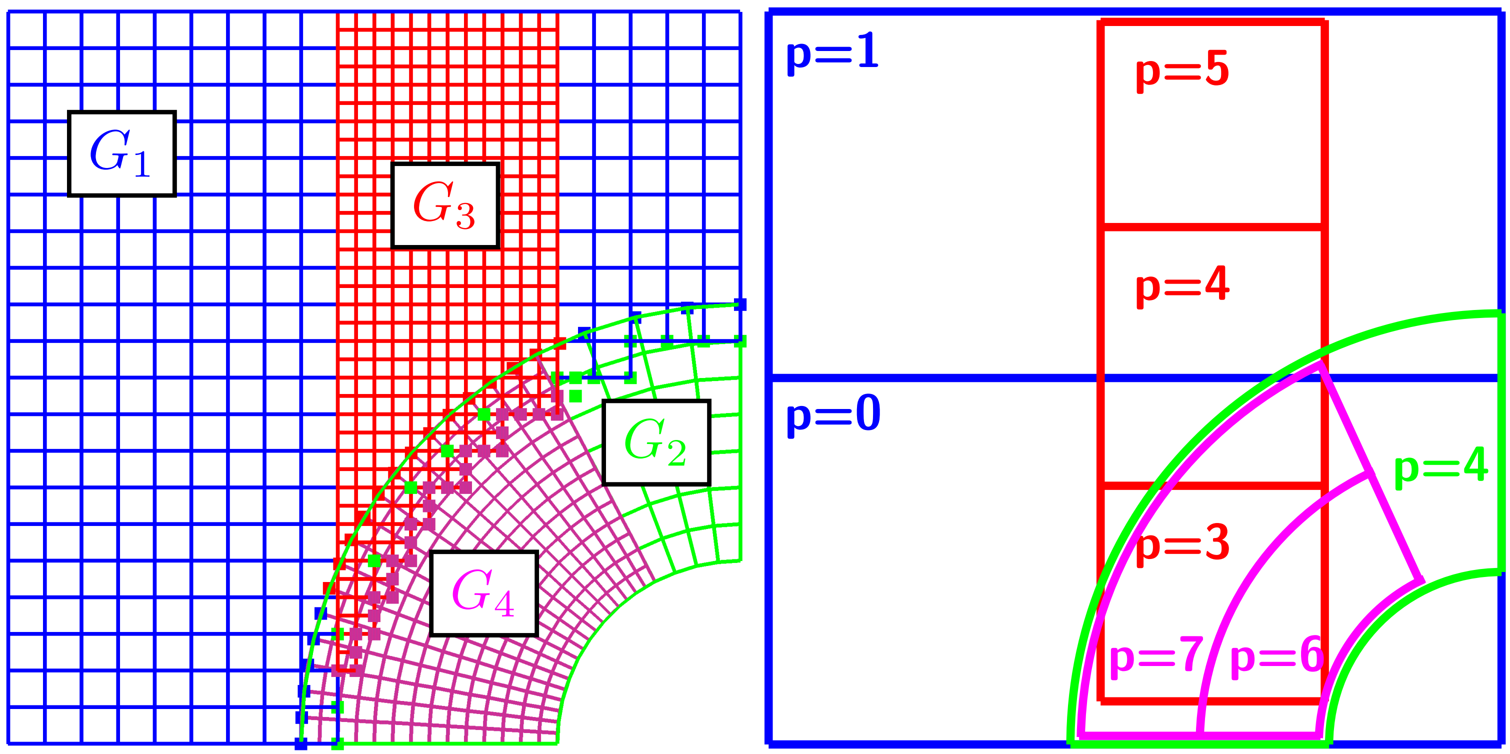
♢ Load balancing: the parallel distribution of the grids is determined using a modified bin-packing algorithm.

Reference:

WDH., D. W. Schwendeman, *Parallel Computation of Three-Dimensional Flows using Overlapping Grids with Adaptive Mesh Refinement*, J. Comp. Phys. **227** (2008).

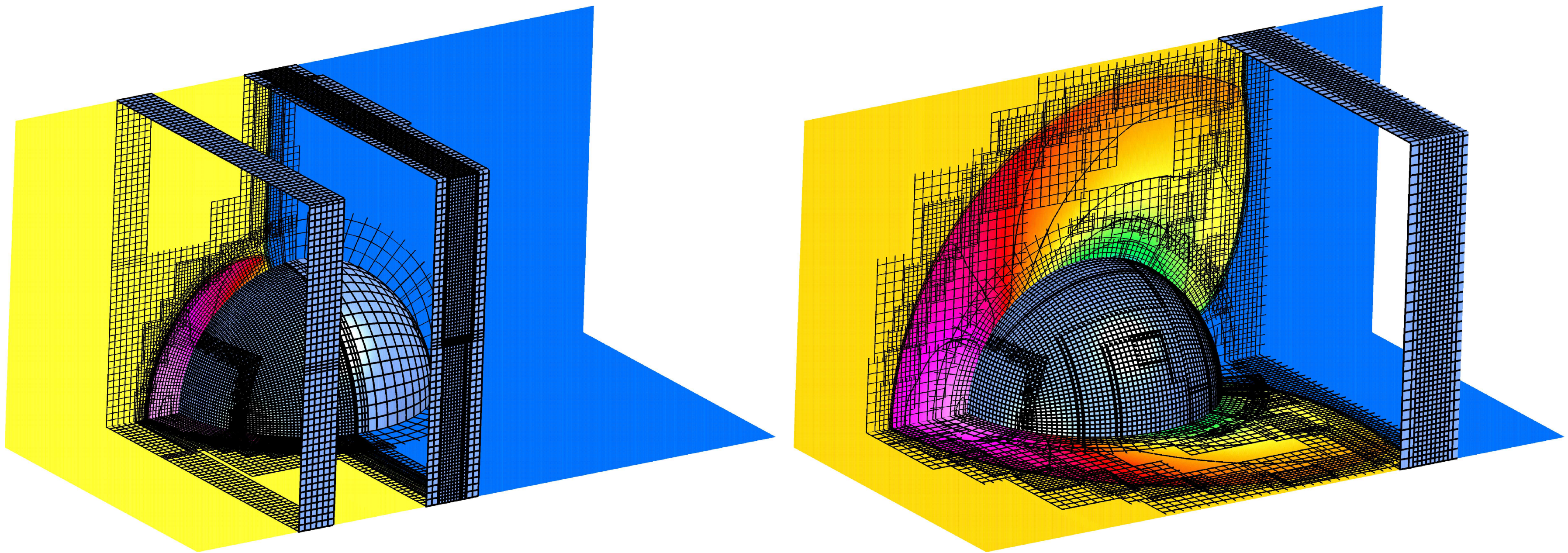
WDH., D. W. Schwendeman, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005).

Distributing Overlapping and AMR grids in Parallel



Each base grid or refinement grid can be distributed over a contiguous range of processors. In this example the base grid G_1 is distributed over processors $[0, 1]$, the base grid G_2 over processor $[4]$, the refinement grid G_3 over processors $[3, 4, 5]$ and the refinement grid G_4 over processors $[6, 7]$.

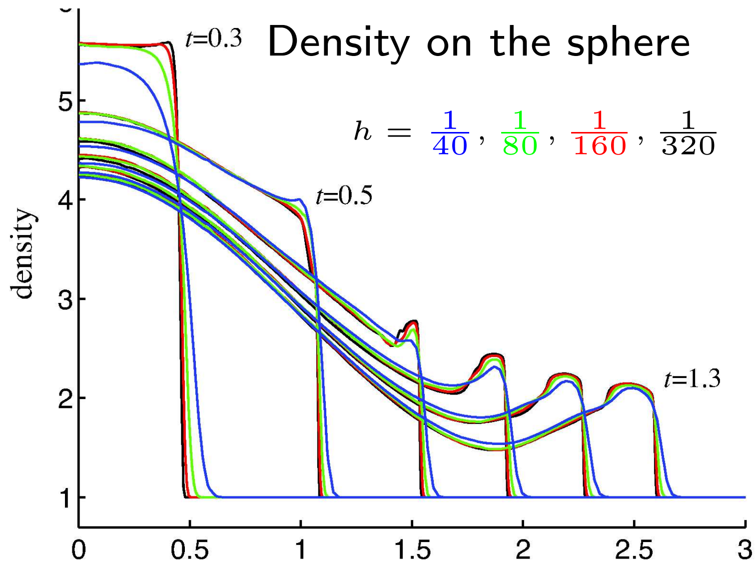
AMR grids for shock diffraction by a quarter sphere

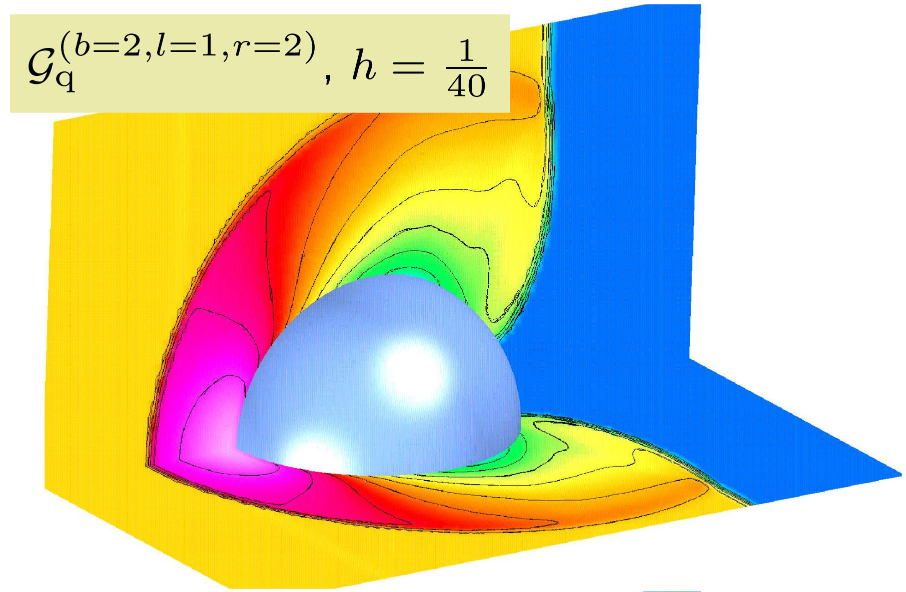


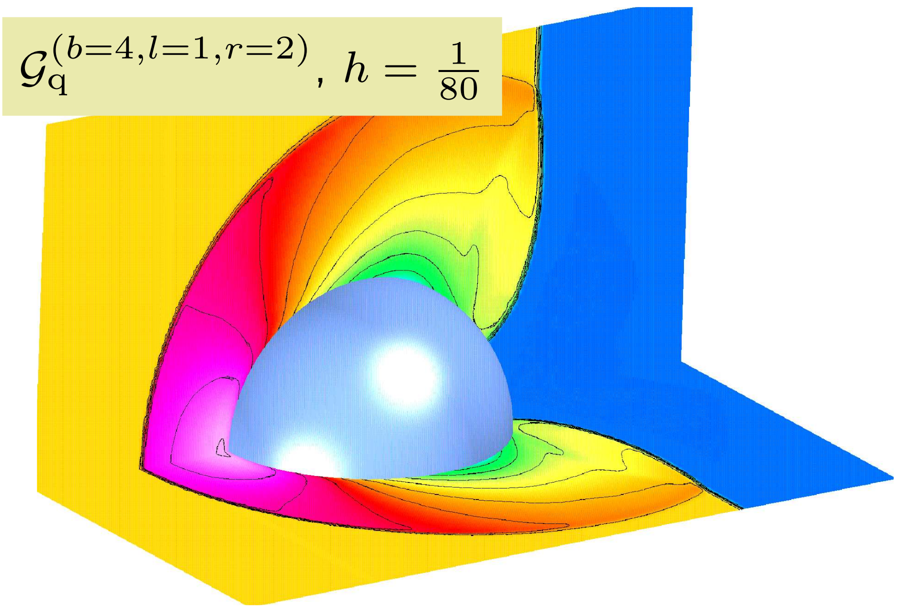
Density and AMR grids for the quarter-sphere problem at $t = 0.6$ (left) and $t = 1.4$ (right). (The grid is coarsened by a factor of 4 for illustrative purposes.)

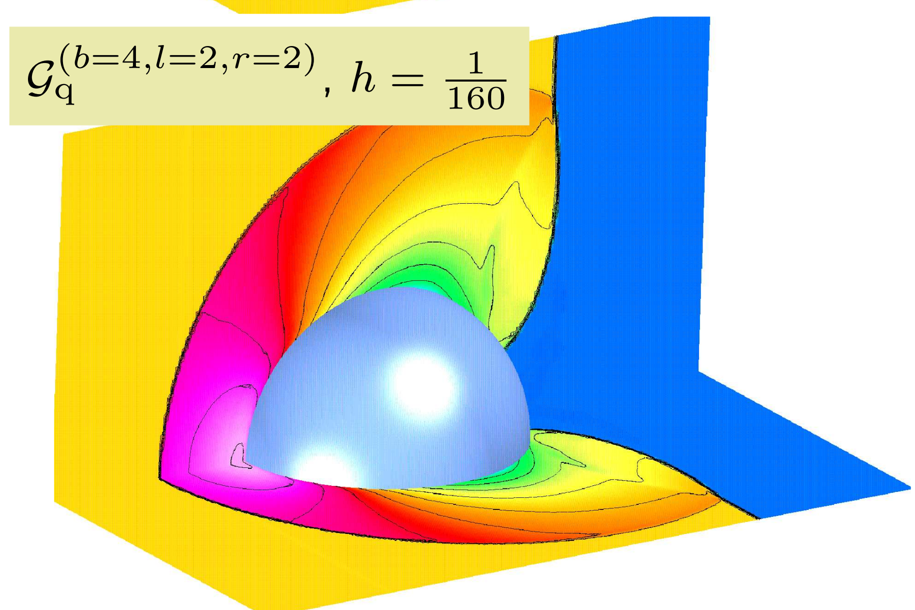
Notes: Euler equations computed with cgcns: two-levels of refinement factor 2, 32 processors, from 6 to 1827 grids, a maximum of 55 million grid points.

Grid convergence study for shock diffraction by a quarter sphere

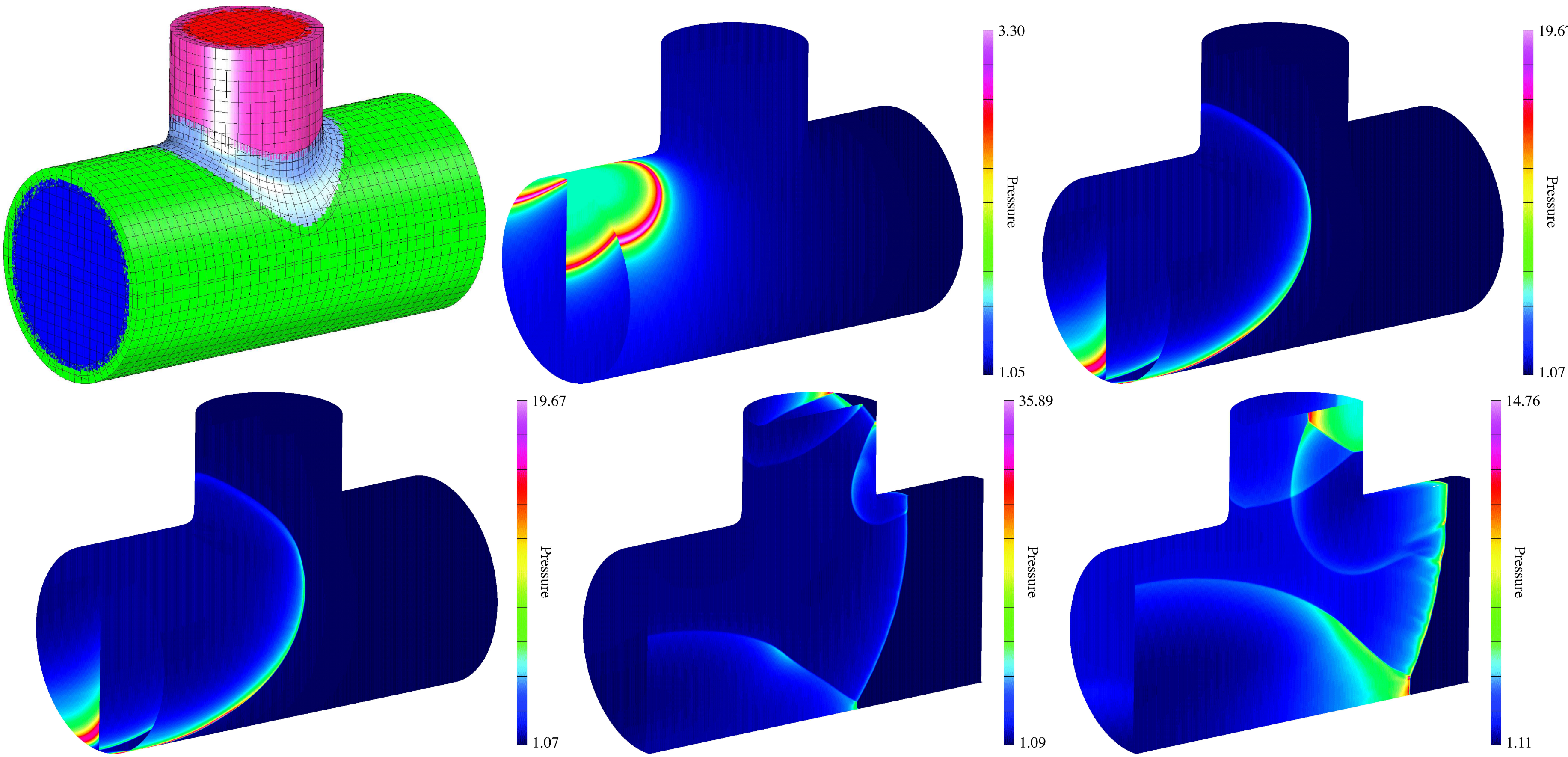






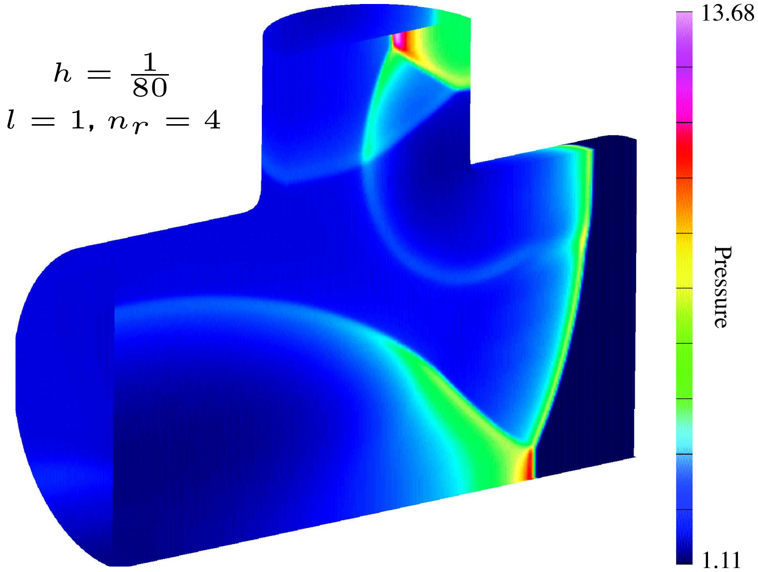


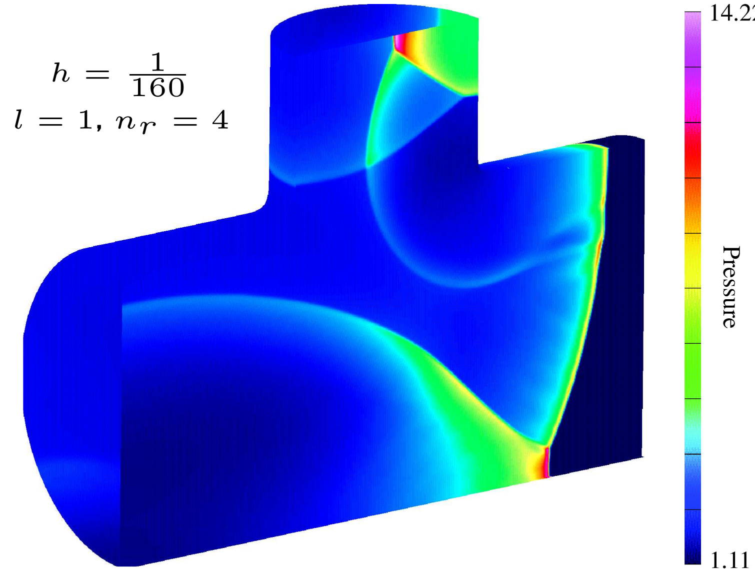
Cgcns parallel AMR example: detonation initiation in a T-shaped pipe

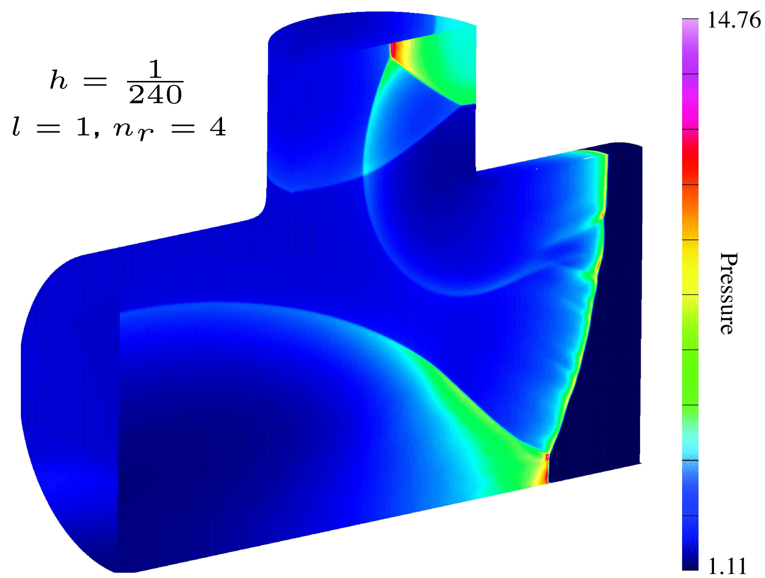


Notes: Reactive-Euler equations computed with cgcns: one level of refinement factor 4, 4930 time steps, 48 processors, from 5 to 682 grids, a maximum of 100 million grid points (effective resolution of 400 million).

Grid convergence study for a detonation in a T-pipe







Estimating Convergence Rates

Define the volume-weighted discrete L_p-norm of a grid function $U_{\mathbf{i}}$ as

$$\|U_{\mathbf{i}}\|_p = \left(\frac{\sum_{\mathbf{i}} |U_{\mathbf{i}}|^p \, d\mathcal{V}_{\mathbf{i}}}{\sum_{\mathbf{i}} d\mathcal{V}_{\mathbf{i}}} \right)^{1/p}, \qquad d\mathcal{V}_{\mathbf{i}} = \left| \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \right|_{\mathbf{i}} dr_1 dr_2 dr_3.$$

We assume to leading order that the discrete solution $U_{\mathbf{i}}^m$ at grid spacing h_m satisfies

$$U_{\mathbf{i}}^m - u(\mathbf{x}_{\mathbf{i}}^m, t) \approx C_{\mathbf{i}}^m h_m^\mu,$$

where $u(\mathbf{x}, t)$ is the exact solution.

Then

$$\|U_{\mathbf{i}}^m - \mathcal{R}_n^m U_{\mathbf{i}}^n\|_p \approx C|h_m^\mu - h_n^\mu|,$$

where $\mathcal{R}_n^m$ is a fine to coarse restriction operator.

Result: Given three solutions we can estimate the convergence rate μ and the error.

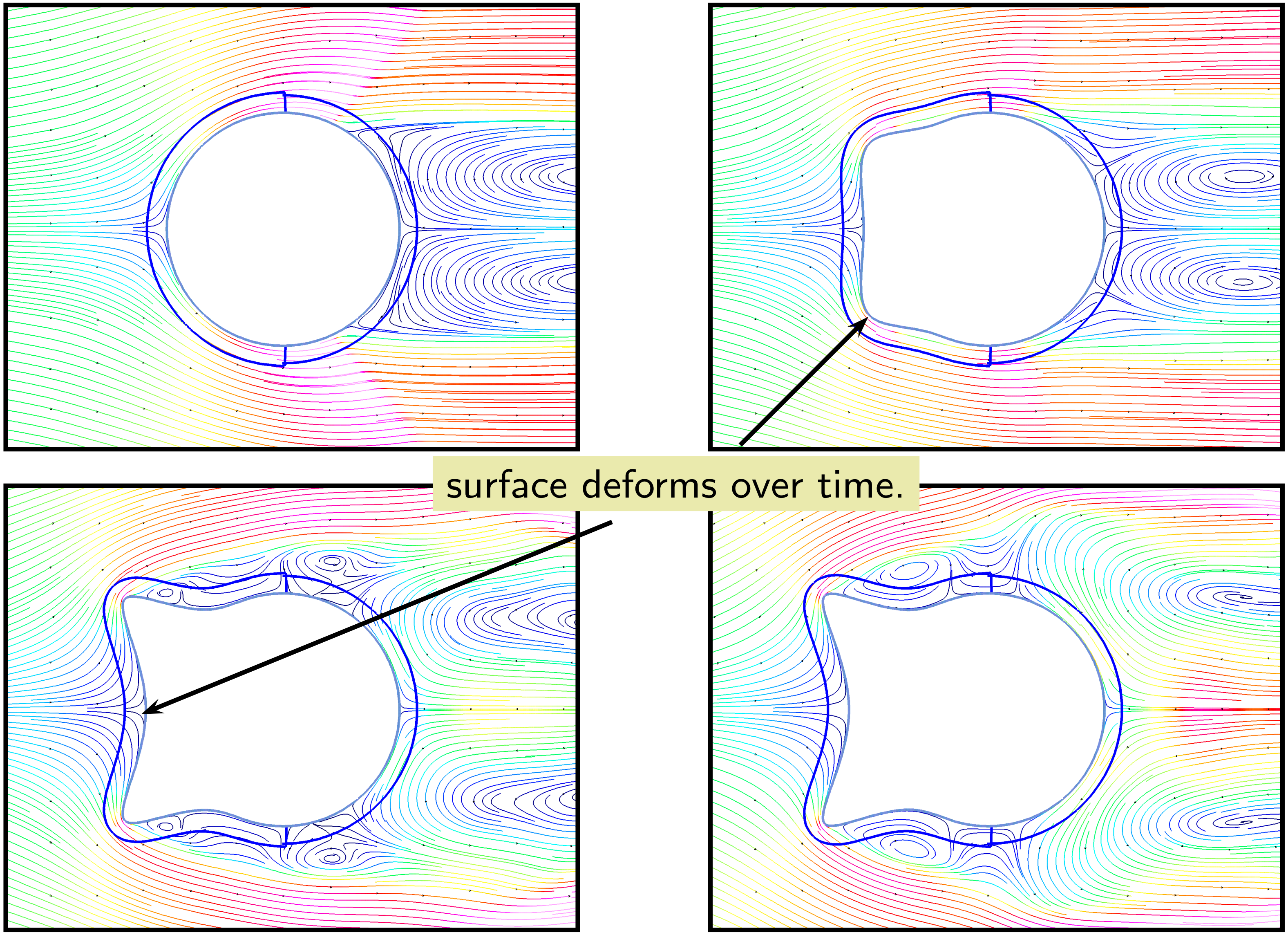
Shock Diffraction by a Quarter-Sphere					
		$t = 1.0$		$t = 1.8$	
Grid	h_m	$\mathcal{E}_1^m$	$\mathcal{E}_2^m$	$\mathcal{E}_1^m$	$\mathcal{E}_2^m$
$\mathcal{G}_{\mathrm{q}}^{(2,1)}$	$1/40$	1.5e−2	8.9e−2	2.3e−2	9.7e−2
$\mathcal{G}_{\mathrm{q}}^{(4,1)}$	$1/80$	6.1e−3	4.6e−2	1.1e−2	5.3e−2
$\mathcal{G}_{\mathrm{q}}^{(4,2)}$	$1/160$	2.5e−3	2.4e−2	5.4e−3	2.9e−2
rate, μ		1.30	0.95	1.06	0.86

Estimated L_1 and L_2 errors in the density, $\mathcal{E}_1^m$ and $\mathcal{E}_2^m$, respectively, and convergence rates μ at $t = 1.0$ and $t = 1.8$.

Detonation in a T-Pipe				
	$t = 2.0$		$t = 2.8$	
h_m	$\mathcal{E}_1^m$	$\mathcal{E}_2^m$	$\mathcal{E}_1^m$	$\mathcal{E}_2^m$
$1/120$	4.0e−3	3.0e−2	3.8e−2	2.6e−1
$1/160$	2.2e−3	1.6e−2	2.4e−2	1.9e−1
$1/240$	9.8e−4	7.1e−3	1.2e−2	1.2e−1
rate, μ	2.04	2.07	1.65	1.09

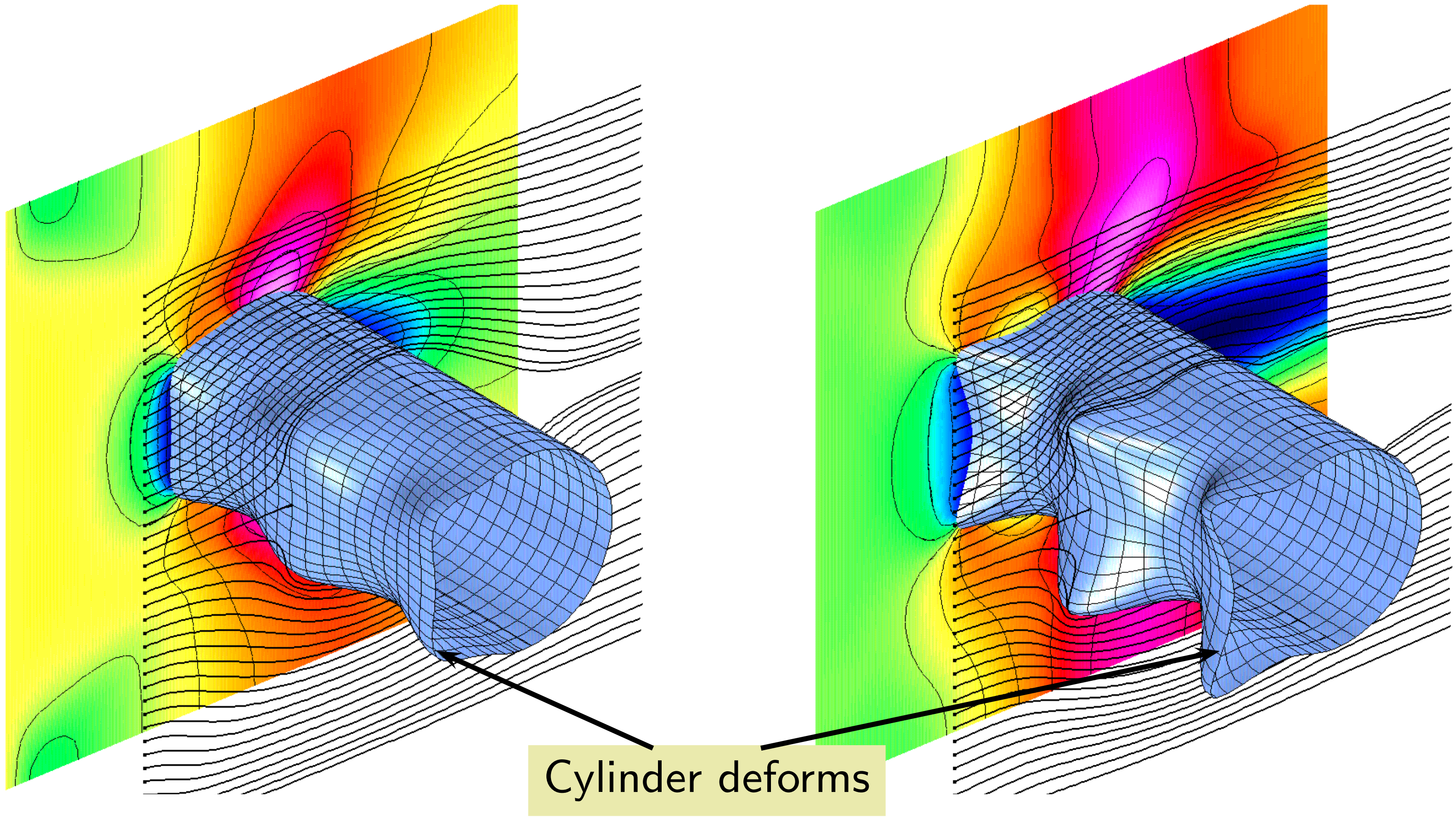
Estimated L_1 and L_2 errors in the density, $\mathcal{E}_1^m$ and $\mathcal{E}_2^m$, respectively, and convergence rates μ at $t = 2.0$ and $t = 2.8$.

Modeling Deforming Geometry with Overlapping Grids



Streamlines of a compressible flow around a deforming boundary.

Deforming Body Applications



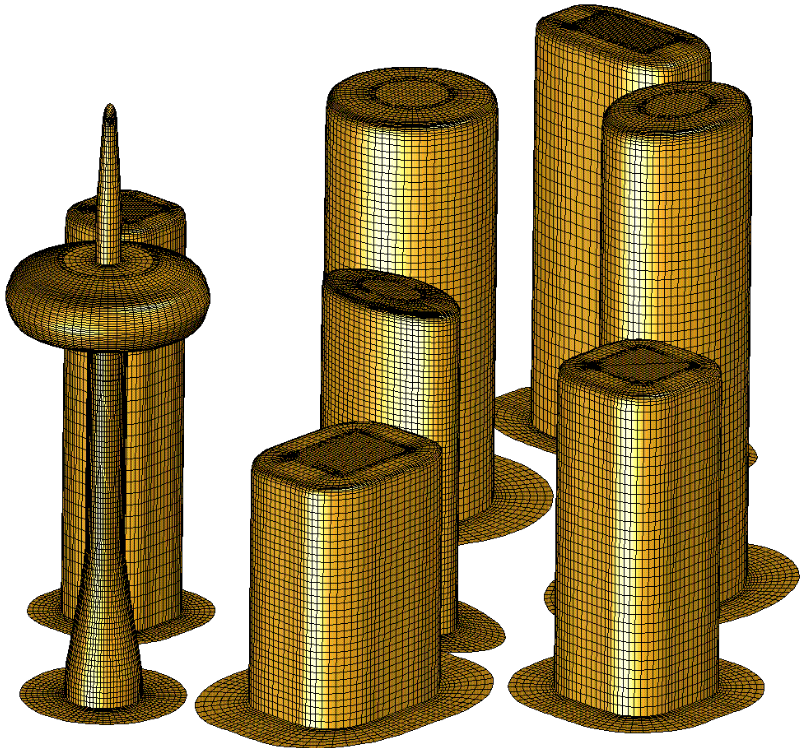
Compressible flow past a deforming cylinder. The surface of the cylinder deforms over time to mimic the growth of ice. Only a subset of the grid lines are shown.

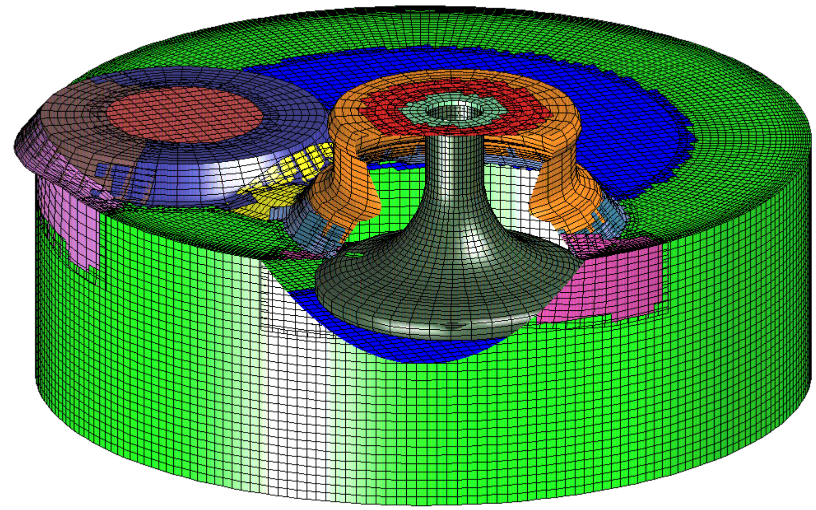
Overlapping Grid Generation

The two main steps in the grid generation process are:

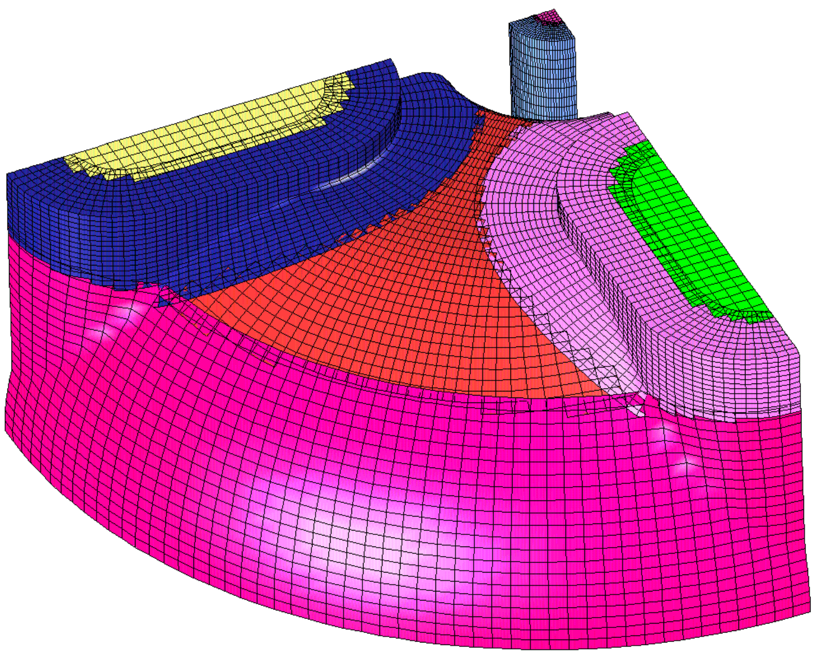
$\diamondsuit$ Build individual component grids (box, sphere, cylinder, NURBS, marching on a CAD or triangulated surface, body-of-revolution, swept-surface, TFI, ...)

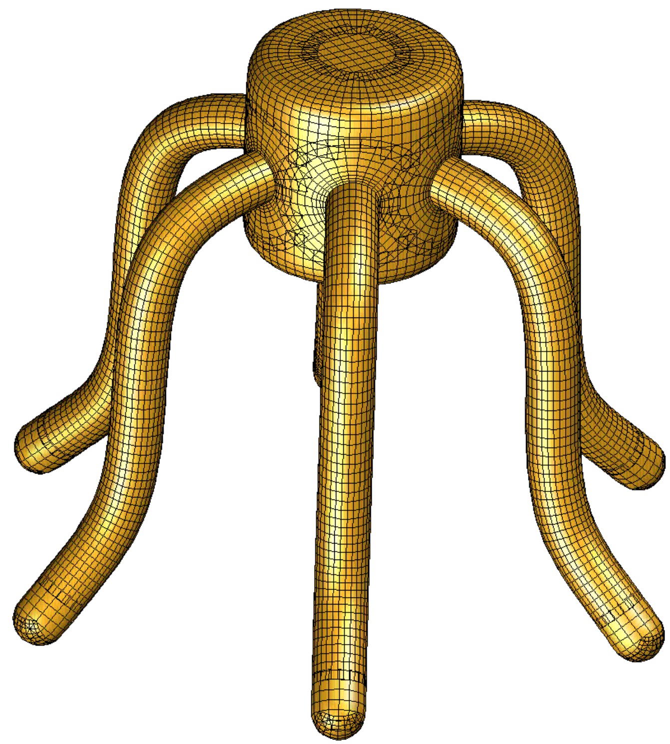
$\diamondsuit$ Construct the overlapping grid: remove un-needed points (*hole-cutting*) and determine interpolation points between the grids.

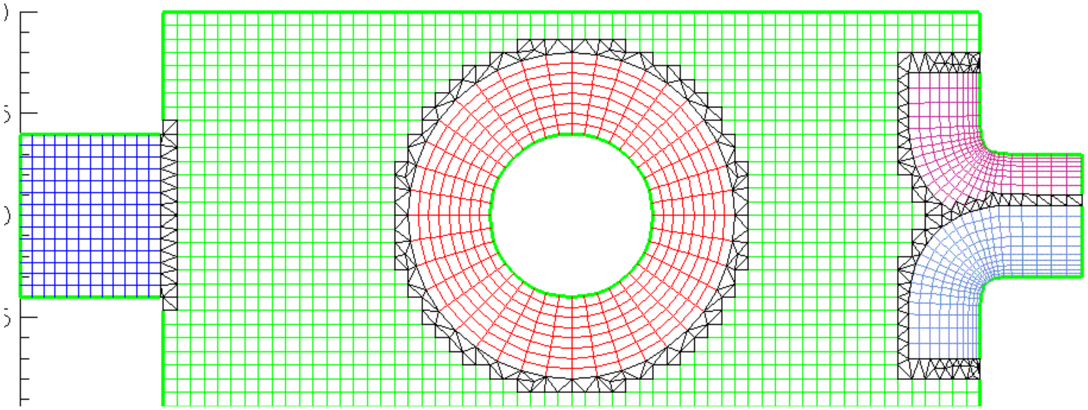


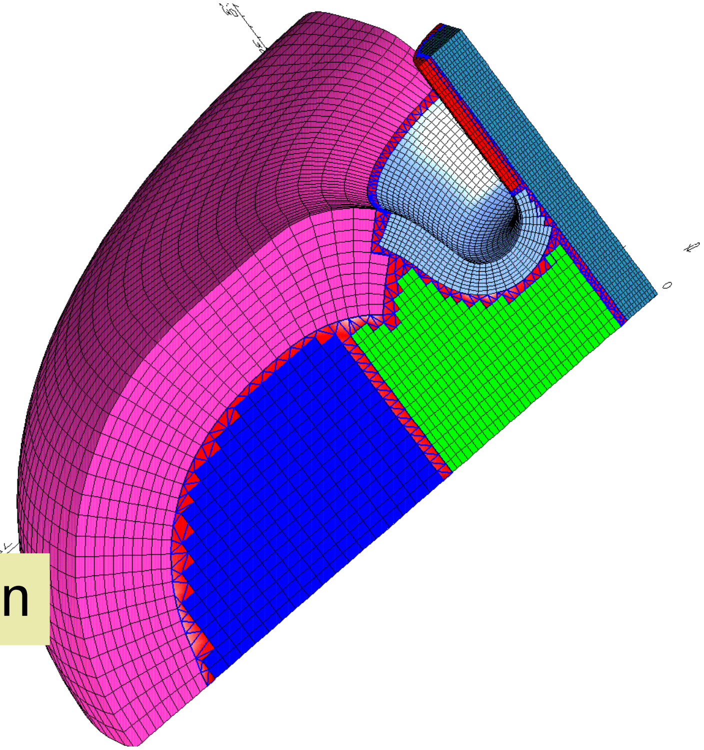


Sample 3D overlapping grids

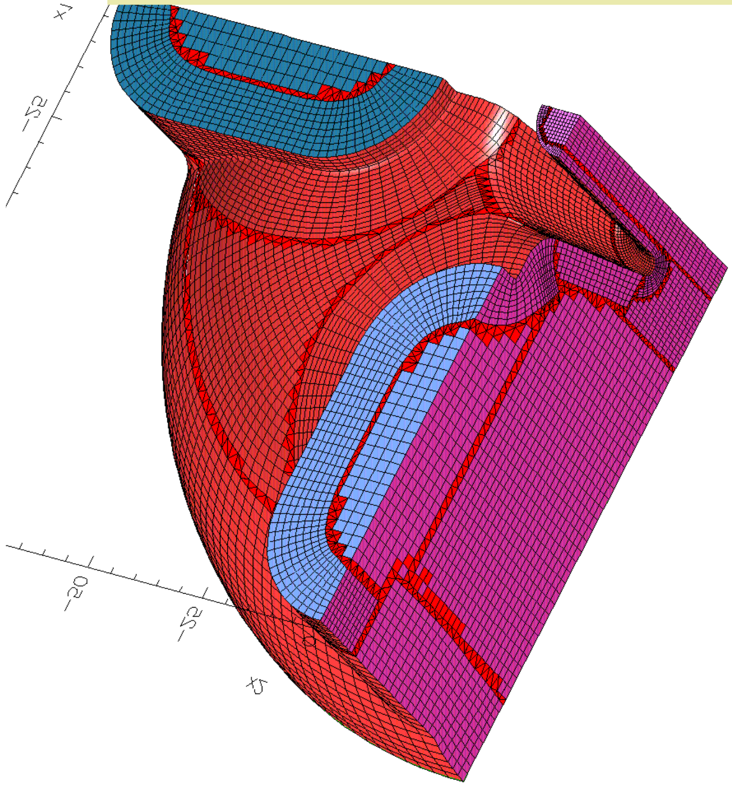


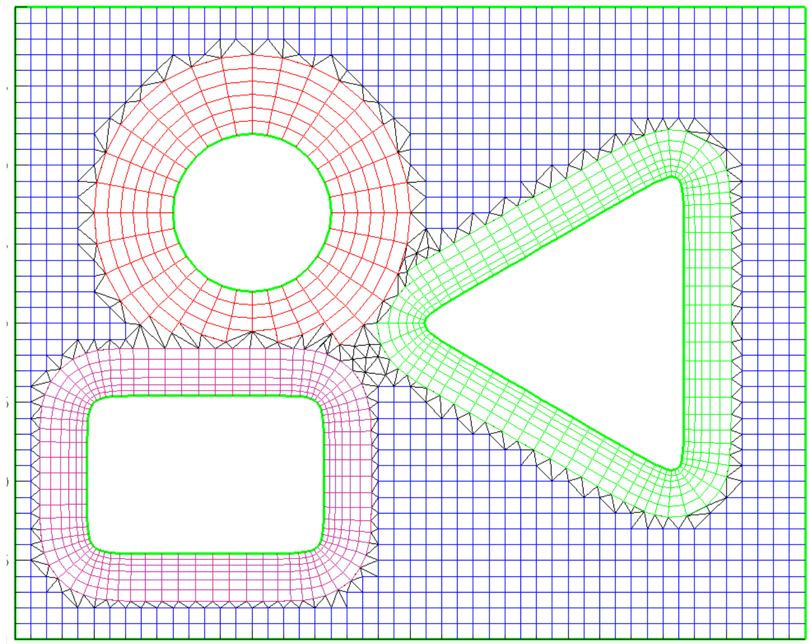






Sample hybrid grids built by Ugen





Ogen: Overlapping Grid Generator

The parallel version of Ogen is now under development (it works in many cases).

Brief description of the algorithm:

- grids are ordered by priority; interpolation is preferred from higher priority grids.
- physical boundaries automatically cut holes in nearby grids.
- robust algorithm with backup rules and interactive error diagnostics.

Brief description of capabilities:

- arbitrary stencil widths supported (single fringe, double fringe, triple fringe ...)
- arbitrary order of interpolation (linear, quadratic, cubic,...)
- optimized for common analytic mappings (spheres, cylinders, NURBS).
- optimized for Cartesian grids – e.g. grid points are not stored when computing the overlapping grid.
- script files with embedded perl commands for "automatic" parameterized grid generation.

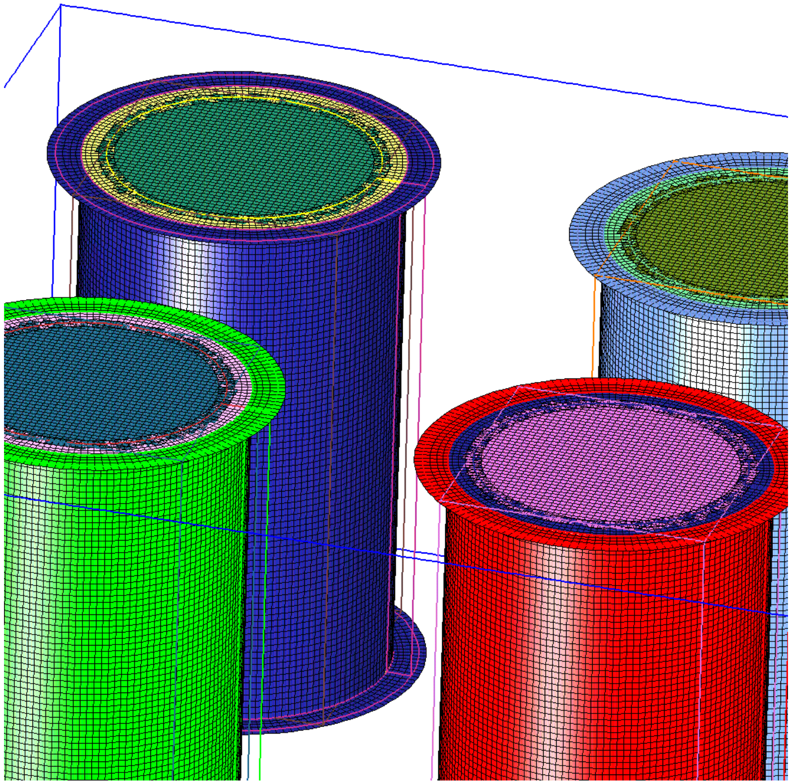
Ogen parallel example: A grid for a multi-domain simulation

♢ A test grid for thermal hydraulics problems: fluid flowing around heated metal rods.

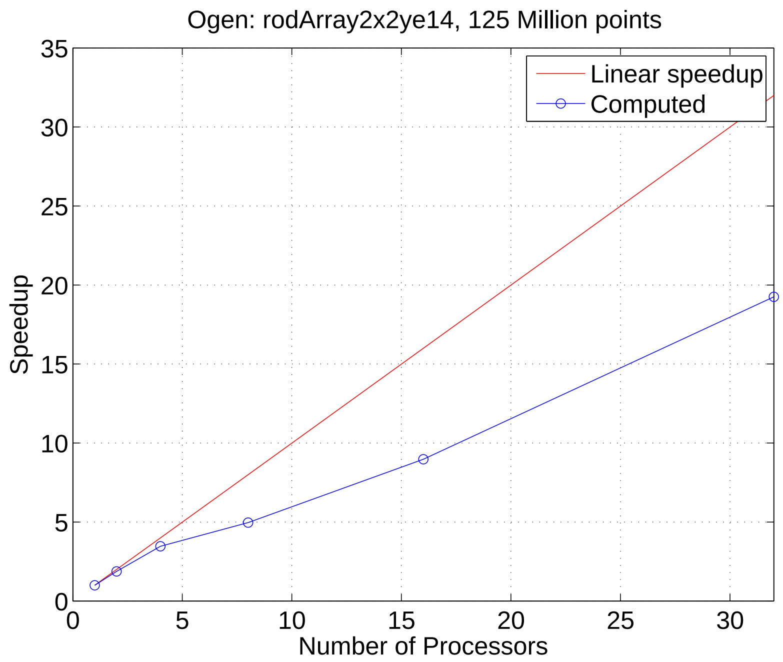
♢ The parameterized ogen script `rodArray.cmd` can be used to automatically generate a grid for an array of N_x by N_y rods.

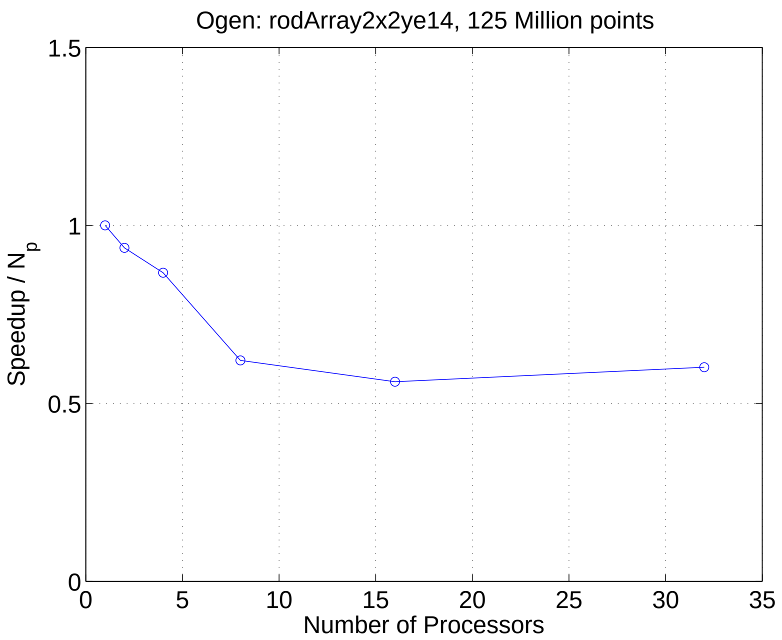
♢ Example: `ogen noplot rodArray -factor=4 -nCylx=2 -nCyly=2 -interp=e`

♢ Example: `mpirun -np 4 ogen noplot rodArray -nCylx=4 -nCyly=4 -interp=e`



Ogen: Very Preliminary Parallel Speedup - Strong Scaling





Parallel speedup, rodArray2x2ye14, 125 Million points, Zeus Linux cluster.

Ogen: Preliminary Parallel Grid Generation Statistics

Grid	N_p	points (M)	cpu (s)	reals/point	points/s/N_p
rodArray2x2ye8	1	24	215	10.5	$1.1e5$
rodArray2x2ye10	1	47	391	9.9	$1.2e5$
rodArray2x2ye12	2	80	341	10.3	$1.2e5$
rodArray2x2ye14	2	125	522	10.0	$1.2e5$
rodArray2x2ye14	32	125	51	15.4	$.77e5$

Some grid generation statistics for various grids. Total number of grid points in millions, total time to generate, memory usage measured as reals-per-grid-point and time to generate measured in points-per-second-per-processor

Cgmx: Solving the Time Domain Maxwell's Equations

Maxwell's equations are solved in second-order form:

$$\epsilon\mu\ \partial_t^2 \mathbf{E} = \Delta \mathbf{E} + \nabla\Big(\nabla \ln\epsilon\ \cdot \mathbf{E}\Big) + \nabla \ln\mu \times \Big(\nabla \times \mathbf{E}\Big) - \nabla(\frac{1}{\epsilon}\rho) - \mu\partial_t \mathbf{J},$$

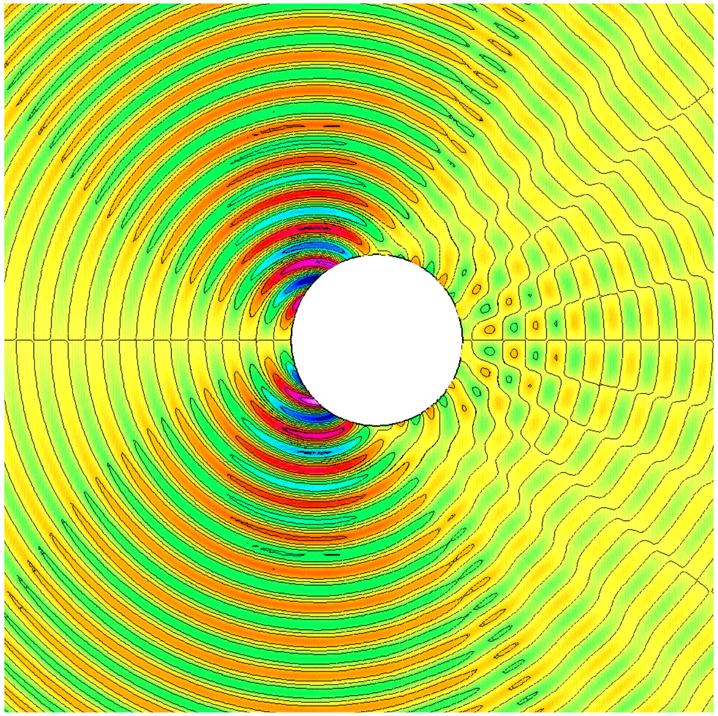
Advantages of the second-order form:

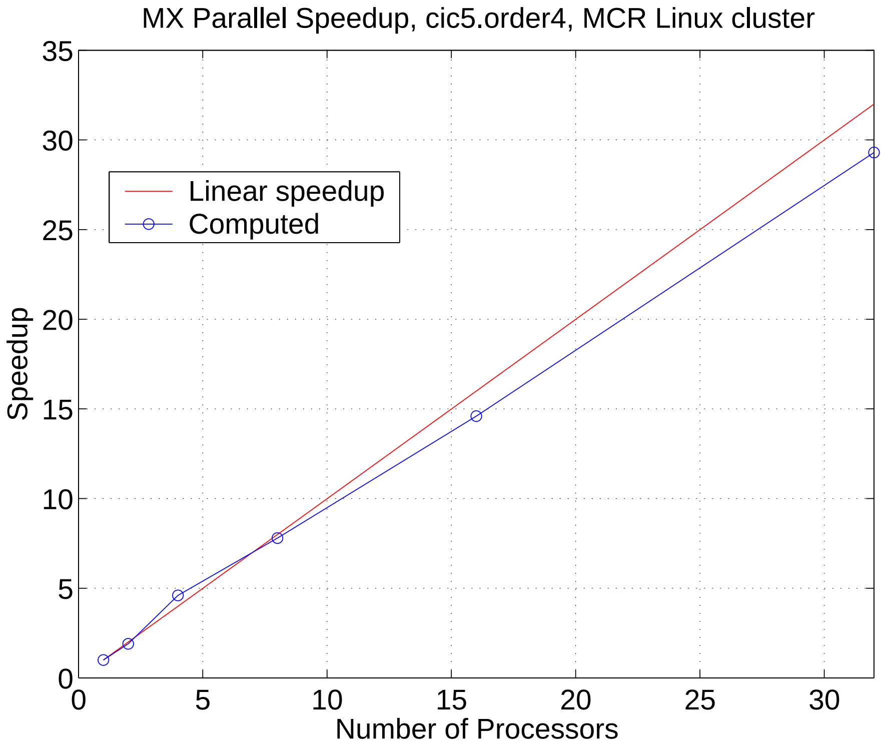
- There is no need to use a staggered grid since the operator Δ is elliptic.
- One can solve for $\mathbf{E}$ alone (and compute $\mathbf{H}$ as a post-processing step).

Features: $\diamondsuit$ 2D, 3D, fourth-order accurate $\diamondsuit$ Efficient time-stepping with the modified-equation approach, allowing a large (cfl=1) time step. $\diamondsuit$ New high-order accurate symmetric difference approximations. $\diamondsuit$ High-order-accurate *centered* boundary and interface conditions. $\diamondsuit$ The scheme is both high-order accurate and nearly as efficient as a Cartesian grid method.

Reference: WDH., *A High-Order Accurate Parallel Solver for Maxwell's Equations on Overlapping Grids*, SIAM J. Scientific Computing, **28**, no. 5, (2006), pp. 1730-1765.

Parallel scaling of the Maxwell solver

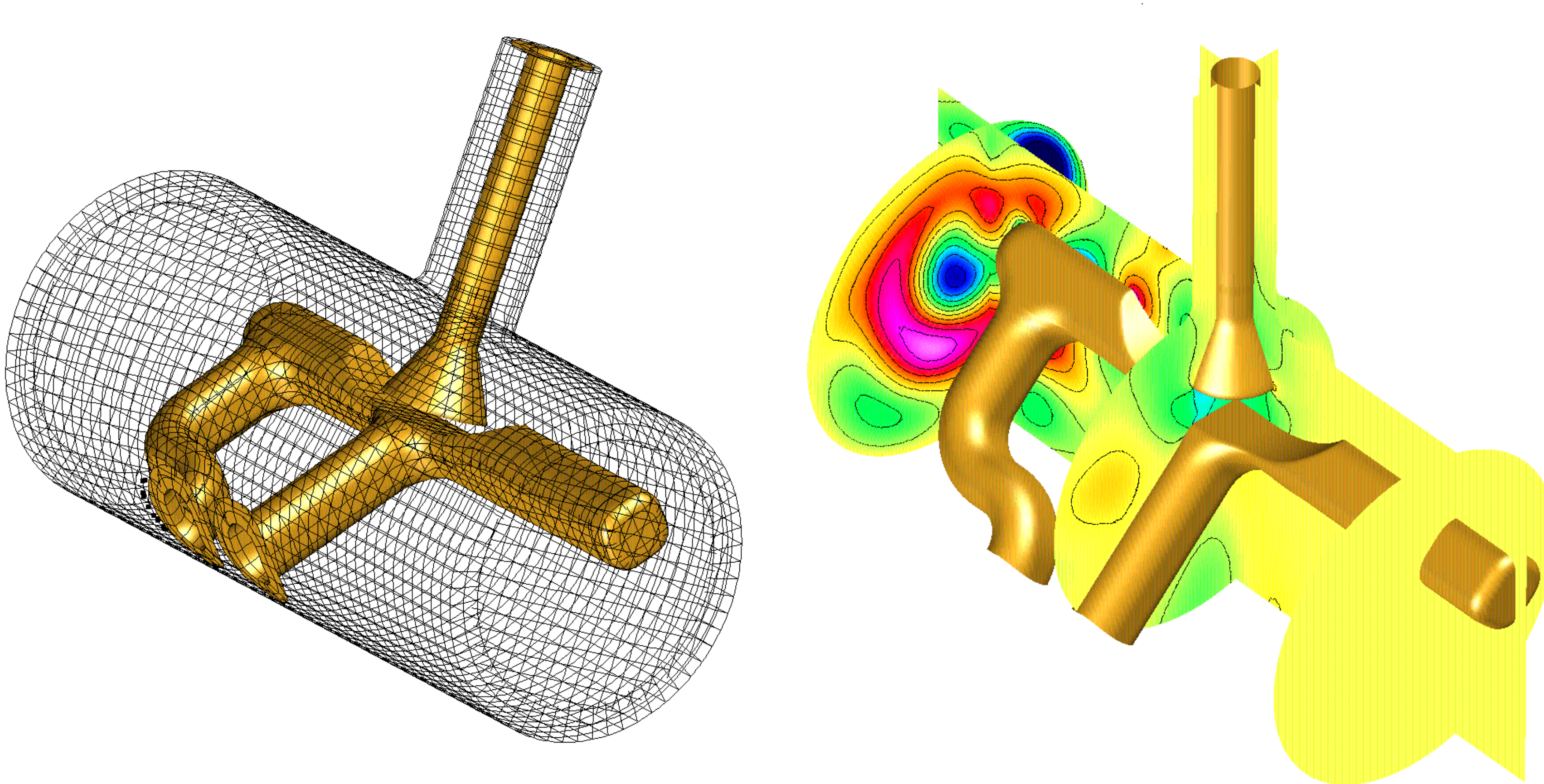




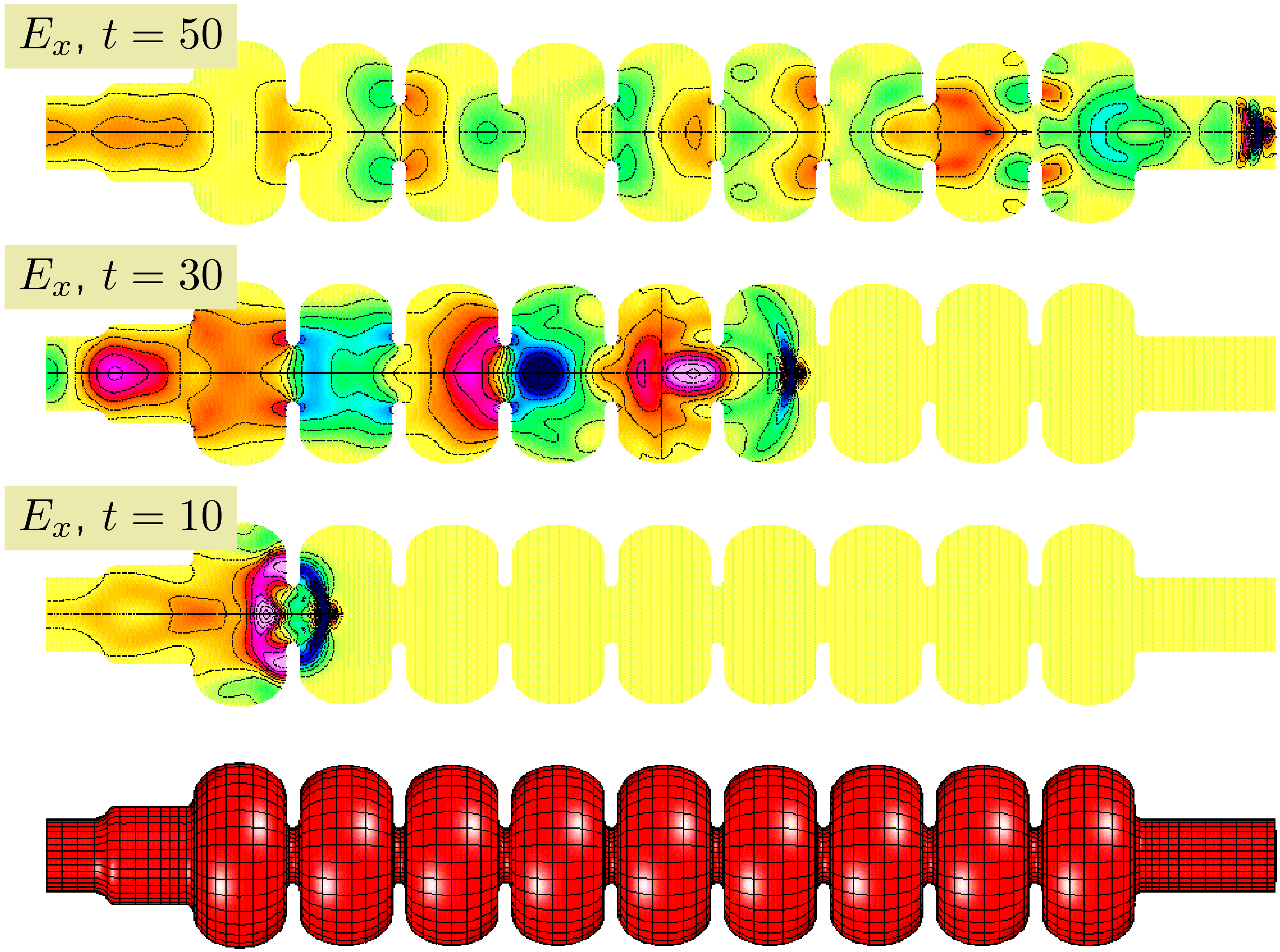
Fourth-order accurate

2D scattering from a cylinder

Fixed size problem, 3.8 million grid-points



An overlapping grid for the HOM coupler and the computed solution (E_x at $t = 2$) for a Gaussian pulse initial condition.



A charge pulse moving through the 3d ILC cavity with no couplers (on a fairly coarse grid, 2nd-order accuracy). A coarsened version of the grid is shown at the bottom.